Applying Clustering and Ensemble Clustering Approaches to Phishing Profiling

John Yearwood, Dean Webb, Liping Ma, Peter Vamplew, Bahadorreza Ofoghi and Andrei Kelarev

Internet Commerce Security Laboratory,
Center for Informatics and Applied Optimization.
University of Ballarat, Ballarat, Australia.
Email: {j.yearwood,d.webb,l.ma,p.vamplew,b.ofoghi,a.kelarev}@ballarat.edu.au

Abstract

This paper describes a novel approach to profiling phishing emails based on the combination of multiple independent clusterings of the email documents. Each clustering is motivated by a natural representation of the emails. A data set of 2048 phishing emails provided by a major Australian financial institution was pre-processed to extract features describing the textual content, hyperlinks and orthographic structure of the emails. Independent clusterings using different techniques were performed on each representation, and these clusterings were then ensembled using a variety of consensus functions. This paper concentrates on using several clustering approaches to determine the most likely number of phishing groups and explores ways in which individual and combined results relate. The approach suggests a number of phishing groups and the structure of the approach can aid the development of profiles based on the individual clusters. The actual profiling is not carried out in this paper.

Keywords: Clustering, Phishing, Graph Partitioning, Cluster ensembles, Profiling, Consensus functions.

1 Introduction

Phishing can be defined as a scam by which an email user is duped into revealing personal or confidential information which the scammer can use illicitly. Phishing attacks use both social engineering and technical subterfuge to steal personal identity data and financial account credentials. Phishing is one of the fastest growing scams on the Internet. The exclusive motivation of phishers is financial gain. Phishers employ a variety of different techniques from spoofed links to malware (keyloggers) to DNS Cache Poisoning (Stewart 2003) (which is also known as Pharming) to lure the unsuspected user into divulging their personal information (Emigh 2005).

A spoofed email is usually sent to a large group of people from an address that appears to be from their bank or some other legitimate institution. The phishing email is typically worded to instil a sense of urgency and to elicit an immediate response from the recipient, e.g., "verify your account details or your account will be closed". The hoax email also contains a link to an online form that is branded to look exactly like the organization website. The form has to be filled in using sensitive information like passwords, user account details and credit card details. Until recently most phishers used the names of financial institutions to deceive people into giving away their account information. They now also use the names of other organizations like eBay, PayPal and even the Australian Taxation Office.

Most technical approaches to phishing so far aim to detect and block or highlight phishing activities either in the original email or when the website is contacted. Examples include the work of Fette, Sadeh and Tomasic (2007), Wu et al (2006), Juels et al (2006), Chandrasekaran et al (2006), Chau (2005), and Jakobsson (2005). For example, the eBay Toolbar is a browser plugin that eBay offers to its customers, primarily to help them keep track of auction sites. The toolbar has a feature called Account Guard that monitors the domain names that users visit and provide warning in the form of a coloured tab on the toolbar. The tab is usually grey but it turns green if the user is on eBay or the PayPal site. It turns red if the user is on a site that is detected as spoofed by eBay. These approaches aim to protect an individual user from the actions of phishers, but they fail to address the issue of protecting the broader community.

Therefore here we propose the development of a complementary set of technology with the aim of profiling the behaviour of phishers, thereby allowing tracking, prediction and possibly identification of these illegal operators.

The rest of the paper is organized as follows: Section 2 gives the background of profiling. Section 3 provides the details of 3 groups of clusterings according to different feature types. Section 4 describes different types of consensus functions. Section 6 shows the evaluation methodologies, and experimental results are shown and discussed in Section 7. Finally Section 8 concludes the work and highlights a direction for the future research.

2 Profiling

'Profiling is a data surveillance technique which is little understood and ill-documented, but increasingly used. It means generating suspects or prospects from within a large population, and involves inferring a set of characteristics of a particular class of person from past experience' (Roger 1993). In (Roger 1993), different data surveillance techniques have been surveyed; like front-end verification and data matching. As well as different problems needing to be tackled in this area, it has been shown that profiling data requires different sets of measures. Take the definition of profiling as in (Roger 1993): 'Profiling is a technique whereby a set of characteristics of a particular class of person is inferred from past experience, and data-holdings are then searched for individuals for close fit to that set of characteristics.' Further-

more numerous potential areas for the use of profiling have been identified. These include patients who have a likelihood of suffering from certain diseases, students having potential artistic talents, identifying customers buying patterns and many others.

Forensic Psychology is used by Webb (2007) to identify perpetrator(s) of a crime based on the nature of the offence committed and its mode of operation (Alison et al. 2003), (Castle and Hensley 2002). This leads to determination of various aspects of criminal psychology before, during and after the crime is committed.

In this paper, we follow the same trend set up by these studies, to profile phishing emails based on their structural characteristics, their content and information about their likely origin. The approach in the first instance is to try to firmly identify the emails that are similar across all of these types of characteristics and assume that these correspond to different phishing groups with certain modus operandi. The next stage of the work (not reported in this paper), will be to construct profiles of these groups by identifying the link structure, orthographic structure and content character of each group.

3 Clustering techniques

The 2048 emails used in our experiments are a subset of a much larger corpus, obtained from a major Australian bank. These are emails gathered by their information security group over a span of 5 months in 2006, and were identified as phishing emails. Most of the emails are about 1026 characters in length and have both text and hyperlink content embedded in them. Some of them contain HTML script, including tables, images, links, and other structures that can be useful in differentiating the emails. Hence, defining the modus operandi of the individual phishing group or activity.

There are a number of features which could be used as a basis for comparison and clustering of these email documents. These include the actual text content displayed to the user, the textual structure of this content, the nature of the hyperlinks embedded within the message, or the use of HTML features such as images, tables and forms.

An approach is to represent each email document in terms of this set of features, and then apply a clustering algorithm to these feature vectors. However, there are two drawbacks to this simplistic approach. Firstly, the nature of the features is varied, some features are numeric whilst others are binary or categorical. Thus combining the features together in a single clustering algorithm is problematic. Secondly, clustering algorithms always produce a set of clusters, even if there is no evidence of any underlying structure in the data. In our case there are no ground-truth labels to use as a basis for testing the clustering results, as the actual source for any of the emails is unknown. Therefore, it is important that methods to validate the clusters produced by the system are found.

Similar issues have previously been observed in the context of clustering of high-dimensional data sets such as those used in bioinformatics. Researchers working in these areas have proposed the use of cluster ensembles. Several independent clusterings are performed based on different subsets of the complete feature vector. These are then combined together in a cluster ensemble to form a final, consensus clustering (Strehl and Ghosh 2002, Topchy et al. 2003, Fern and Brodley 2004). If pairs of objects (i.e. emails) are observed to be commonly grouped together across all of the independent clusterings, this provides increased confidence that the clustering indicates a genuine relationship between the objects rather than just random noise. In (Fern and Brodley 2004) the different feature sets used in each clustering were random subselections or projections of the original feature vector. In our approach, we have instead chosen to use three groups of features which reflect different characteristics of the underlying data:

- the text content which is shown to the email's reader
- a characterisation of the hyperlinks in the email
- the orthographic features of the email

3.1 Text clustering

Perhaps the most obvious feature to use in profiling emails is the textual content displayed to the reader. For the emails in the data set this textual content was encoded in a number of ways. They included plain text, as HTML-formatted text, or as an embedded image. Therefore pre-processing was necessary to extract the text content from each email, by stripping away HTML tags and other structural information and by applying optical character recognition to the embedded images.

With the text extracted, this was then converted into a numerical feature vector for each email by computing the TF/IDF weight algorithm, shown in equation (1).

$$w_{ij} = tf \cdot idf = f_{ti} \cdot \frac{\log(N)}{df_{\boldsymbol{k}}} \tag{1}$$

n = total number of emails
f_{ti} = frequency of term i in document k
df_k = number documents containing di

Each email was then represented as a vector of its term weights and these vectors were clustered using the k-means algorithm.

3.2 Hyperlink clustering

We grouped together emails based on similar tokens found within the fake hyperlink structure. Many of these hyperlinks contained similar names, this was especially apparent for their directory naming conventions. We looked for directory or file names that were obscure, frequently occurring and had names that were related to banking. We ensured that no legitimate directories or file names were included, as all legitimate bank hyperlinks were removed before the clustering process. All non legitimate and frequently occurring bank related names/tokens were used as a seed for each cluster. Any emails containing absent links, links with no directories, IP only or hex only links or links containing none of these key tokens were clustered together into the "other" cluster. Shown in Table 1 are the directory and file names used to build each cluster. The following is a more detailed description of the hyperlink clustering procedure.

3.2.1 Extraction of links

1. Firstly, we extracted all links from the 2048 emails.
2. We then pre-processed the links by removing all surrounding tags and script information and any other periphery not directly related to the file or name structure of the link itself.

3. Next we removed all legitimate links belonging to any bank.
4. The links were then broken up into their Protocol, Host name, multi level domains and multi level directory components. All protocol, host name and top level domain name identifiers were removed during this process. These include such things as "http:","www","edu","au" and all others.
5. All remaining tokens found in the second level domain and all other directory levels were then stored in a binary tree along with the number of emails they were found in and their overall corpus frequency count.
6. Any emails that had no tokens, due to absent links, legitimate bank links only or had no words present in the link, were automatically clustered into cluster 10, "Others" and didn't contribute further to the clustering process.

3.2.2 Building the clusters

This was a partially manual process where we looked over the stored tokens, taking into account the names used, their overall frequency and the number of emails that they appeared in. With the possibility of such a large number of words, we ignored all words with a frequency of 1% of the total number of emails, and were not bank related. The exception however was the "moreinfo.html" and "wumoreinfo.html" file name. Words that may of occurred many times, but were found only in a few emails were also excluded as a grouping could not be formed from them. Large frequency words such as "index", "netbank" or "bigpond", were excluded as they were not unique enough to belong to just one group. However names such as "index2_files", "nabib" and ".verify"were kept due to their higher obscurity and number of emails they were found in. Some examples are:

nabib

- http://blog.co.tz/**nabib**/
- https://ib.national.com.au/**nabib**/help/
- http://startherefilms.com/**nabib**/
- http://evolk.info/**nabib**/
- http://floridanetservices.com/**nabib**/
- http://www.jr.ac.th/**nabib**/

/r1/?

- http://www.netbank.commbank.com.au.netbank.rim2s.biz/**r1**/c/
- http://citibusinessonline.da-us.citibank.com.dllinfo.tv/**r1**/cb/
- http://www.barclays.co.uk.customercare.goto.confpr.st/**r1**/b/
- http://www.national.com.au.vdq6270z.manicte.com/**r1**/n/
- http://www.barclays.co.uk.customercare.goto.mabberas.com/**r1**/b/

Table 1: Number of emails in initial groups found

Hyperlink Keyword	Number of emails
/nabib	289
/.verifyacc/ , /.ver/,/.verify/	22
/index2_files/	98
/anb2/	6
/r1/?/	765
/netbank/ or /netbank/bankmain/	319
cbusol	41
/.national.com.au	81
moreinfo.htm,wumoreinfo.htm	8
"Others"	419

With the seeding tokens now found, all remaining emails not belonging to the "Others" group were then allocated to the cluster containing their nominated token.

Shown in table 1 are the nominated fake link tokens (Cluster seeds) as well as the number of emails contained within each cluster. The "Others" row shows the remaining number of emails that didn't contain viable hyperlink tokens.

3.3 Orthographic clustering

Phishing emails usually contain multimedia type information to help overcome phishing filters and lure the unsuspecting recipient. This includes images and text, where the text information may contain plain text, markup languages and styles, scripts, URLs and so on. The images may contain logos or a mock up of a bank or an institution's web page with altered text. However, the information cannot be recognized by a system directly, rather it needs to be characterized according to the needs of the system.

Phishing emails are largely similar in content. Therefore, we believe that orthographic features are important in such an application. The orthographic features mainly consist of style characteristics that are used to convey the role of words, sentences or sections that describe the email content.

Since an email body is often loose in structure, parsing email content is more difficult than parsing the header part of the email. For the present we have defined the features manually based on observation. The orthographic features collected in our system are described as the following:

1. size of the text and html body of an email.
2. whether an email has text content[1].
3. number of visible links in an email.
4. whether a visual link is directed to the same hyperlink in an email.
5. whether an email contains a greeting line.
6. whether an email contains a signature at the end.
7. whether an email contains HTML content.
8. whether an email contains scripts.
9. whether an email contains tables.
10. number of images in an email.
11. number of hyperlinks in an email.

[1]Some phishing emails contain only images

12. whether an email contains a form.

13. number of fake tags in a email[2].

A high level description of the feature extraction and clustering process can be seen in Figure 3. Features are collected according to features defined above, but not all the features are informative. Therefore, the most informative features are selected using a learning model and clustering is carried out. Both of these tasks are done iteratively using the Modified Global k-means algorithm (Bagirov and Mardaneh 2006, Bagirov 2008). A selection process conducts a search for a best feature subset and then uses Modified Global k-means (MGkm) for the evaluation of the current feature subset. This is run repeatedly on the phishing emails using various feature subsets and various tolerance values for MGkm. The performance is evaluated by MGkm objective function values on the various feature subsets, where the subset with the lowest objective function value is chosen as the iterated feature subset on which the induction algorithm runs.

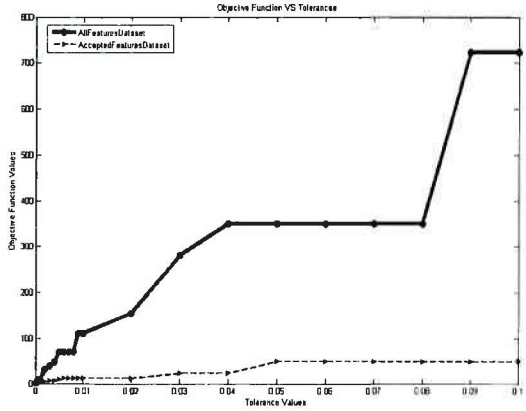

Figure 1: Objective function values vs tolerance values

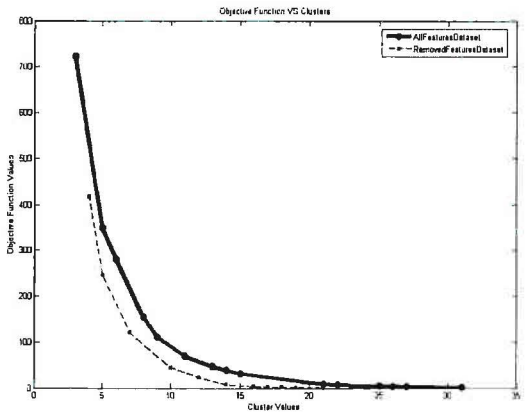

Figure 2: Objective function values vs the number of clusters

Figure 1 shows the relationship between the objective function values(V_{of}) and tolerance values (V_t)

[2]Sometimes phishers use ill formed HTML or embedded fake tags in an attempt to elude phishing filters

We calculated V_{of} over a range of V_t from 0 to 0.1 and discovered V_{of} achieves stable value of 45 when $V_t \geq 0.05$. Figure 2 illustrates the relationship between V_{of} and the number of clusters. The graph shows that when V_{of} is 45, the number of clusters is 9. Together these figures indicate that a good clustering (a good balance between objective function values, tolerance and the number of clusters) of this data set is achieved with 9 clusters.

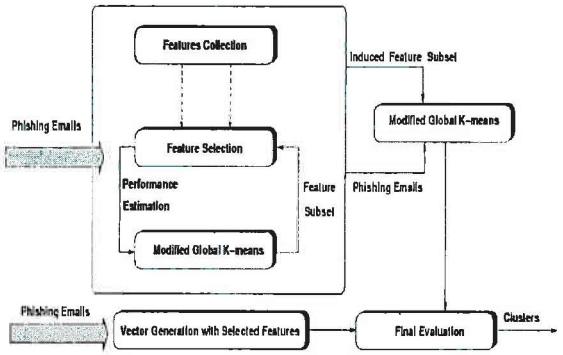

Figure 3: The feature selection and clustering process using orthographic features

4 Consensus functions

Several consensus functions have been proposed for forming consensus clusterings from an ensemble of independent clusterings (Strehl and Ghosh 2002, Topchy et al. 2003, Fern and Brodley 2004). Given a data set $X = \{x_1, x_2, ..., x_n\}$ where n is the total number of instances x_l, $\Pi = \{\pi^1, \pi^2, ..., \pi^r\}$ is a clustering ensemble on X where r is the total number of clusterings and $\pi^i = \{c_1^i, c_2^i, ..., c_{K_i}^i\}$ where c_j^i corresponds to the cluster j in the clustering π^i and K_i is the total number of clusters formed in the clustering π^i. For each π^i we have $\cup_k c_k^i = X$.

Consensus clusterings are usually used on a single data set with different clusterings produced by:

- the different subsets of the whole feature set, or
- the different initial parameters in some clustering algorithms.

In this work, we use consensus functions on different clusterings obtained using the different features already discussed in Section 3. We utilize four consensus functions described by Fern and Brodley (2004).

- *Instance-Based Graph Formulation (IBGF)*: This method constructs a graph in which instances are represented by nodes and their connections are modelled as weighted edges given the association between the instances. The weight on the edge between the instances x_l and x_m in IBGF is calculated using the formula in equation (2).

$$
\begin{aligned}
w(l,m) &= \frac{1}{r}\sum_{q=1}^{r} I(g_q(x_l) = g_q(x_m)) \\
I(arg) &= 1;\ \textit{if arg=true} \\
I(arg) &= 0;\ \textit{if arg=false} \\
g_q(arg) &= c_k^i;\ \textit{where } arg \in c_k^i
\end{aligned}
\tag{2}
$$

IBGF makes use of a graph partitioning algorithm[3] to partition the graph according to the edge weights. The final clustering includes clusters corresponding to each graph partition.

- *Cluster-Based Graph Formulation (CBGF)*: This method constructs a graph with the clusters as graph nodes and the similarity of the clusters as weights on the edges. The edge weight between two clusters c_i and c_j in CBGF is calculated using the Jaccard Measure in equation (3).

$$w(i,j) = \frac{|c_i \cap c_j|}{|c_i \cup c_j|} \tag{3}$$

  A graph partitioning algorithm is then used to eliminate the lowest weighted edges, thereby ensuring that clusters which share a large number of instances will be grouped together in the final consensus clustering. Following partitioning, each instance is assigned to the final cluster in which it most commonly occurs.

- *Hybrid Bipartite Graph Formulation (HBGF)*: This method constructs a bipartite graph with two types of nodes, clusters and instances. There is an edge between each pair of nodes; however, the weights of the edges between the nodes of the same type are 0. The edge weight between an instance x_l and a cluster c_i is 1 if $x_l \in c_i$ and is 0 otherwise[4]. The graph partitioning algorithm partitions both clusters and instances simultaneously and the final clustering is formed according to the partitions of instances.

- *K-Means Clustering Function (KMCF)*: This method was first proposed by Topchy et al. (2003) and uses the standard K-Means clustering algorithm to produce the final clustering. KMCF first creates a set of new features for each clustering π^i. It adds K^i binary features to the new set of features. Each of the new features correspond to a cluster in π^i. The total number of new features is equal to the total number of the clusters in Π. For each instance $x_l \in X$, the feature corresponding to c_i is 1 if $x_l \in c_i$ and is 0 otherwise. The new features are standardize to have zero mean and then the standard K-Means is applied to the features to create the final clustering.

5 Background to the experiments

Described in Section 3 are the text, links and orthographic structural clustering techniques. Each technique individually assigned each instance of an email to a cluster or profile according to its clustering criteria and feature set. Hence, capturing specific but different aspects of the data, where a single clustering technique alone could not. Our aim therefore is to combine these clusterings together to

- reinforce the intersecting information
- include information not shared between the three techniques
- find the best fitting number of profiles.

[3] In our implementations, we utilize the METIS graph partitioning module developed by Karypis and Kumar (1998).

[4] We set the weights between same node types to 1 and if an instance belongs to a cluster we set the weight to 1000. This is because of the implementation limitations in METIS where weights cannot be 0.

Shown in Section 4 is an explanation of the four consensus functions CBGF, HBGF, IBGF and KMCF. In terms of their application most work to date has been done on cluster ensembling of large data sets. The main focus here has been in breaking large data sets into smaller subsets using such techniques as random projections or random sub-sampling. Then, clustering algorithms could work on the smaller more manageable subsets of the original data set. The consensus functions would then be used to recombine these multiple clusterings so that a final clustering could be found. This work is shown in (Strehl and Ghosh 2002, Fern and Brodley 2004), where a comparison of all the consensus functions is undertaken on multiple data sets. Their results show that on average the HBGF and IBGF consensus functions improved the most when increasing both the number of projections and projections within the ensembles. The other two consensus functions performed less favourably under these conditions. However, these results were also dependent on the data set, and both the CBGF and KMCF consensus functions performed better than the other two on certain data sets.

Our application of these consensus functions is very different. We use different feature sets and clustering algorithms to find the clusterings. We also use a much smaller number in our ensemble, using only three technique clusterings. In (Fern and Brodley 2004) consensus functions are being compared in the use of many applications. One such application is Robust Centralized Clustering (RCC). Here, they look at a fixed number of clusterings; they have access to the dataset's features and use ten different but diverse clustering algorithms. This application is very similar to ours. The authors also show that a CBGF type consensus function performed very well in such an application. Unfortunately the KMCF consensus function was not examined in this study.

The culmination of the clusterings by the three techniques into one final clustering via the four consensus functions leaves us with at least four final clusterings. However, the number of cluster labels given to the consensus functions from the individual clustering techniques could vary between any or all of them. For example, in our case both the text and link clustering techniques have ten cluster labels, whereas the structural orthographic technique has nine. The consensus functions do not automatically determine what final number of cluster labels is the most appropriate. This means that we must specify the number of cluster labels for the final consensus clustering results.

From our previous examination of the fake links represented in the emails, an approximation of ten profiles was found. Furthermore, results from the structural orthographic technique shown in Section 3 using the Global Modified k-means algorithm (Bagirov 2008) reports nine clusters as an optimal number of clusters. We assumed then, that around 10 clusters could best partition the data set in terms of the number of profiles. With that in mind we set about establishing this final approximate number of clusters. In doing so, we casted a wider net by giving the consensus functions a range of 5 to 15 clusterings. This would allow us to evaluate five final cluster configurations either side of our initial assumption.

6 Evaluation Criteria

Evaluating the best final consensus clustering from 5 to 15 was a non-trivial process. To give us an indication of the "most correct" final clustering we employed the use of three measures, these were, Normalized Mutual information(NMI), purity and the number of edge cuts. We compared each final consensus

clustering 5 to 15 to each of the individual technique clusterings by comparing their intersections and relative information using the NMI and purity measures. We also compared the number of edge cuts given by the consensus functions from each final clustering. We surmised that the best final clustering would have the following:

1. A relatively consistent NMI value close to 1 when comparing the three given individual clustering techniques against all of the final consensus clusterings.
2. A high but consistent two way purity value. That is, a value similar when comparing both the individual technique to the final clustering and vice versa. We again would expect to have a value close to 1, to show that there is a strong intersection between both the individual technique clustering and the final consensus clustering.
3. A relatively low number of edge cuts given the number of clusters. This value is compared to all other clusterings within its respective consensus function as well as within the other consensus functions.

6.1 Purity

Purity measures the quality of a clustering solution by determining the number of points in the intersection of allocated clusters and predetermined labelled classes.

Let k be the number of clusters found by a hard partitioning clustering algorithm in data set D. Let $|c_j|$ be the size of cluster c_j and $|c_j|_{class} = i$ be the number of points of class i assigned to cluster j. Then the purity of cluster j is given by

$$purity(c_j) = \frac{1}{|C_j|} \max_i(|c_j|_{class=i}) \quad (4)$$

The overall purity for cluster $c_j, j = 1, \ldots, k$, is expressed as a weighted sum of the k individual purities

$$purity(c_j)_{tot} = \sum_{i=1}^{k} \frac{|c_j|}{|D|} purity(c_j) \quad (5)$$

However, the purity measure shown here is asymmetrical. Let $|class_i|$ be the size of class c_i and $|class_i|_{clust} = j$ be the number of points of cluster j assigned to class i. Find $purity(c_i)$ and $purity(c_i)_{tot}$. Then $purity(c_j)_{tot} \neq purity(c_i)_{tot}$ unless the points are symmetrically distributed between both the respective class i and cluster j.

It is therefore a relative measure that depends on the order in which the multi labelled set of instances are measured. Since we are the ones attempting to label these instances we assume that we do not have the actual classification labels. We therefore, take $purity(technique_{pj})_{tot}$ of clustering technique p where p is an integer mapped to each technique type "links", "text" and "structural orthographic" against final consensus function m where $m = 1, \ldots, 15$, the number of clusters found by each respective consensus function. We then find $purity(consensus_{mi})_{tot}$ against each $technique_p$ and $purity(technique_{pj})_{tot}$ against $consensus_m$. Leaving us with two purity measures comparing a two way symmetric intersection between the respective final consensus and individual technique clustering. Allowing us to measure the difference between the two. Hence, the smaller the distance between the two purity measures, the better the intersection between the two clusterings.

6.2 Normalized Mutual information

Mutual Information is a symmetrical measure that takes into account both the intersection of the two sets of clusterings as well as quantifying the statistical information found in both distributions, see (Cover and Thomas 1991). Though it provides a good indication of the shared information between a pair of clusterings, it is desirable as with purity, to have a normalized version of Mutual Information with values between $[0 - 1]$.

Let X and Y be random variables described by the consensus function clusterings $\lambda^{(i)}$ and the technique clusterings $\lambda^{(j)}$ where $i = 1 \ldots p$ and $j = 1 \ldots m$, with $k^{(i)}$ and $k^{(j)}$ number of clusters respectively. Let $I(X, Y)$ denote mutual information between two random variables X and Y and $H(X)$ and $H(Y)$ denote the entropies of both variable X and Y respectively. In the literature several normalizations exist. We chose the version of NMI found in (Strehl and Ghosh 2002, Fern and Brodley 2004) as it has been shown to successfully measure consensus functions against various types of ensembles. It uses the geometric mean of $H(X)$ and $H(Y)$ to normalize the mutual information see (Strehl and Ghosh 2002, Fern and Brodley 2004) for a detailed description and a proof.

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \quad (6)$$

The NMI measure gives a best result when the value is close to one. This happens when the intersection of both $\lambda^{(i)}$ and $\lambda^{(j)}$ is strong and both entropies $H(X)$ and $H(Y)$ have similar values. Thus, NMI is a very good measure as it shows how much information has been preserved and how closely the clusters overlap between the final consensus results compared to the individual technique clusters.

An average or maximum of NMI values were used in (Strehl and Ghosh 2002, Fern and Brodley 2004) across the cluster ensembles created by the random projections or sub-sampling when compared against the final consensus clusterings. Since we have only three techniques this evaluation would be of no advantage to us. Therefore we decided to compare each individual technique against all of the final consensus clusterings individually, using a balance measure between the two purities and NMI to guide us to the best consensus clustering.

6.3 Number of edge-cuts

The Metis Graph partitioning software is used to create the partitions for the consensus functions shown in Section 4. The algorithm used, found in (Karypis and Kumar 1998), computes a k-way partition of a graph by minimizing the number of edge-cuts subject to a number of vertex balancing constraints. The edge-cut value is the total number of edges being cut in order to obtain that final number of clusterings.

We use this measure by dividing the number of edge-cuts by the number final clusters given to the consensus functions. As we have a range 5 to 15 final clusterings, we would expect the number of edge-cuts/number of final clusters to decrease across the 16 clusterings. It is natural that when asking the consensus function algorithm to produce a larger number of final clusterings, it would then make more cuts in order to create more partitions. However, because the algorithm works on minimizing the number of cuts, a number of cuts that is much greater than the previous cluster's cut would indicate a stronger cohesion amongst that partition.

We are then looking for a value that is considerably smaller than the clusters around it, as this shows that the number of cuts has decreased significantly.

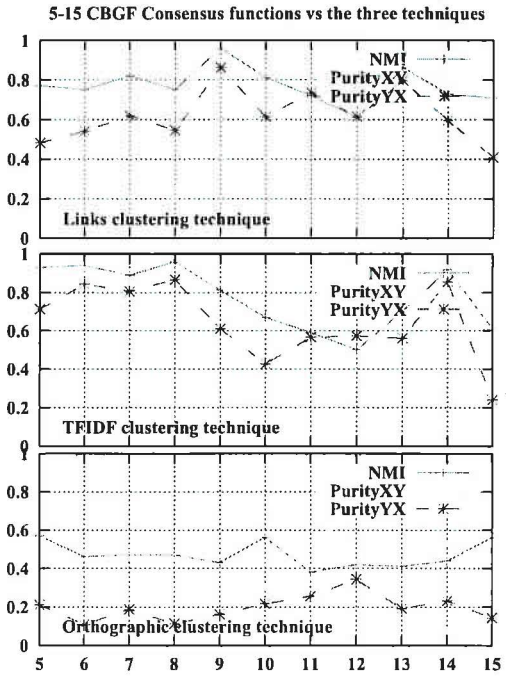

Figure 4: CBGF Purity and NMI measures

7 Results

7.1 Comparing the individual techniques

We compared each individual clustering against one another to find their NMI, purity and individual entropy. As shown in Table 2, the three techniques contained roughly the same amount of information as they all displayed an entropy of between $0.62 - 0.74$.

Table 3 shows the results of comparing the links and text content clusterings using both the NMI and purity measures. It can be seen that the links and text content clustering has the strongest intersection, as all the NMI and purity values are high. Leading us to the conclusion that both the links and text clustering techniques have captured similar information from the data set. However, Table 3 also shows that the structural orthographic clustering result compared to the other two techniques gave a much poorer NMI value. Furthermore, Table 3 also shows a big gap between the two orthographic purity measures, as well as these values being small. It is also worth noting that the results from comparing the Orthographic technique to the other two techniques were comparatively similar. This leads us to the conclusion that the structural orthographic technique has captured mostly different information compared to the other two techniques. This may also be the reason why, when comparing the three techniques to the final consensus clusterings that the Orthographic technique, again had poorer results compared to the other techniques.

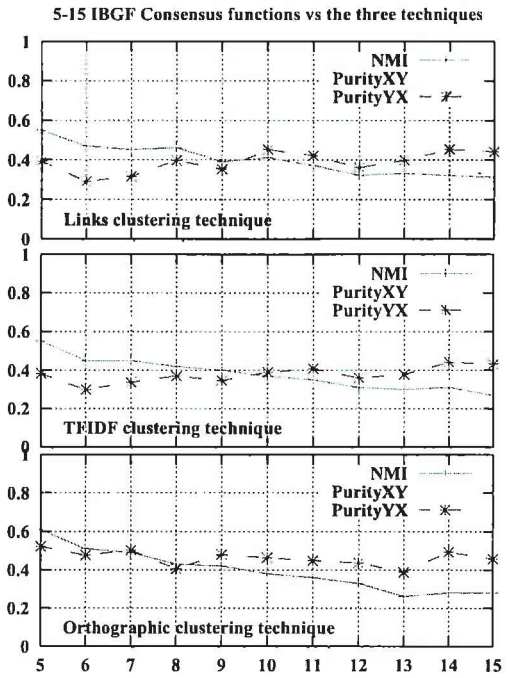

Figure 5: IBGF Purity and NMI measures

Table 2: Entropy of the individual clustering techniques

Technique	H(X)
links	0.737
Text content	0.674
Structural orthographic	0.615

7.2 Comparison of consensus functions

Figures 4, 5, 6 and 7 show the results of NMI and the purity values for each final consensus cluster 5 to 15 compared to each of the three techniques for the four consensus functions. When comparing the results shown in these graphs we can see a lot of variation in each of the different consensus functions. The CBGF consensus function shown in Figure 4 and KMCF consensus function shown in Figure 7 report the best results. They show consistently higher NMI and purity values when compared to the IBGF consensus function shown in Figure 5 and HBGF consensus function shown in Figure 6. The other noticeable difference is that there is much less variation in the difference in the purity values of the CBGF and KMCF graphs compared to the IBGF and HBGF graphs.

At closer inspection, we see that both CBGF and

Table 3: NMI and purity results from comparing clusters techniques against one another

Technique	NMI	Pur(X,Y)	Pur(Y,X)
links vs text	0.77	0.7754	0.6104
links vs ortho	0.41	0.5103	0.1674
text vs ortho	0.46	0.4893	0.1268

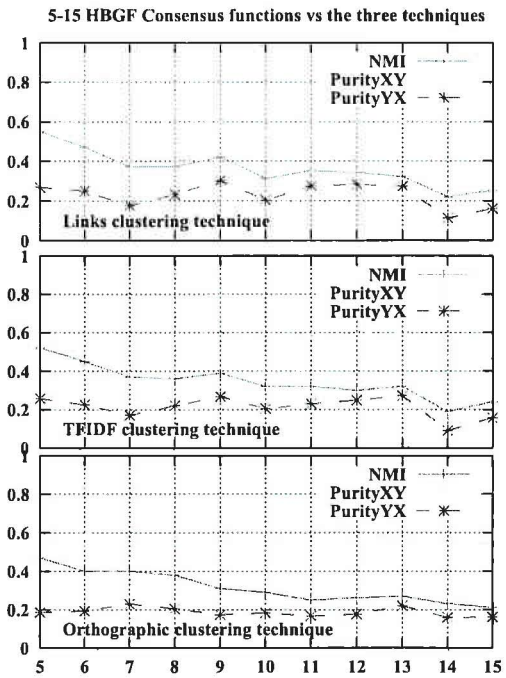

Figure 6: HBGF Purity and NMI measures

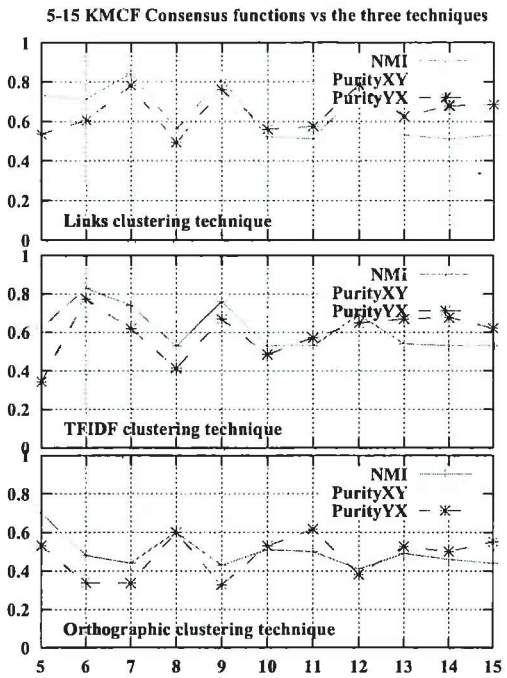

Figure 7: KMCF Purity and NMI measures

KMCF consensus function graphs show the highest NMI values for both the links and text content clustering techniques compared to results shown for the structural orthographic technique. Both the IBGF and HBGF consensus functions, shown in Figures 5 and 6 respectively report the worse results. Both consensus functions give the smallest NMI values compared to the CBGF and KMCF consensus function results. They also show that there are bigger differences in the two purity values again compared to the CBGF and KMCF consensus graph results.

Based on the results from these graphs we can rule out any of the final consensus clusterings produced by the IBGF and HBGF consensus functions. This leaves us with the clusterings obtained by the CBGF and KMCF consensus functions. When comparing the results of both the CBGF and KMCF consensus functions shown in Figures 4 and 7 respectively, we can see that the NMI values given in the CBGF consensus graph for both links and text/TFIDF techniques gives higher values at their respective peaks. The only exception is shown in the orthographic technique graph of the KMCF consensus function. The results shown for both the CBGF and KMCF consensus functions are favourable and warrant further exploring.

7.3 Evaluating the best final clustering

We can utilize both the NMI and purity measures to evaluate the best individual clustering. An ideal result for the final clustering would indicate a NMI value close to 1, both purity values would also be close to 1, but with a similar value. Balanced purity shown in equation (7) fulfils this criteria. It gives a value output in the range of 0 to 3 for each individual clustering technique, where 3 would be the best possible intersection and 0 the least. Our aim therefore, would be to find the largest balanced purity value given across the three clustering techniques for each of the 5 to 15 final consensus clusterings of each consensus function.

We take the sum over the three clustering techniques for each of the 5 to 15 individual clusterings respectively. We then find the maximum value across all of the 5 to 15 clusterings for all four consensus functions, in order to find the overall maximum value. This maximum value would then give the best number of clusters for the best consensus function technique that would in theory best capture our data set.

We denote $purity(c_i)_{tot}$ as $pur(c_i)$ and $purity(c_j)_{tot}$ as $pur(c_j)$, refer to equation (5).

$$\text{balanced purity} = (NMI - |pur(c_i) - pur(c_j)| + pur(c_i) + pur(c_j)) \quad (7)$$

Figure 9 and Figure 10 show the results of equation (7) on both the CBGF and KMCF consensus

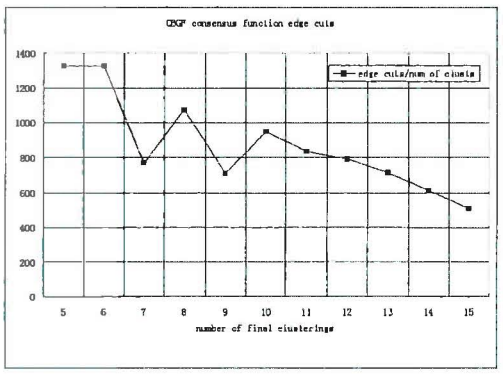

Figure 8: CBGF consensus function number of edge cuts/number of clusterings

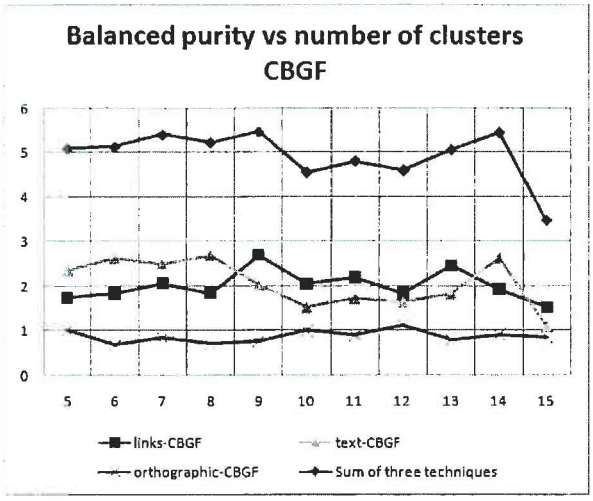

Figure 9: CBGF consensus function sum and individual balance purity measures for the links, text and orthographic techniques

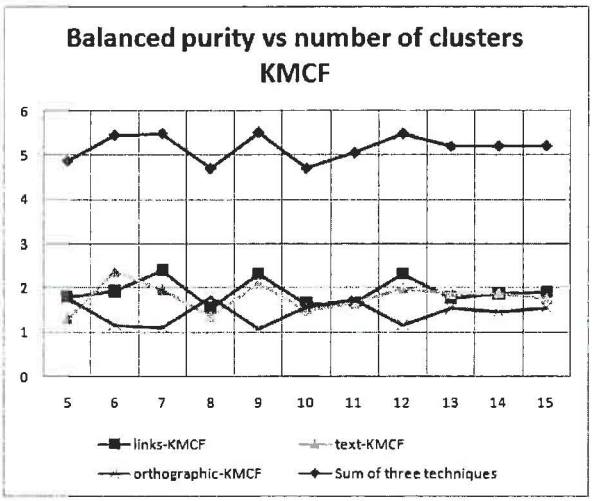

Figure 10: KMCF consensus function sum and individual balance purity measures for the links, text and orthographic techniques

functions against the three clustering techniques. The top values in the graph are the summation of the lower three sets of values that correspond to each of the individual clustering techniques. It can be seen, that in both of these graphs that the largest value found from the summation on the three techniques was the final clustering 9.

Figure 8 shows the number of cuts divided by the number of final consensus clusterings for the CBGF consensus function. As mentioned earlier, you would expect a decreasing graph with little fluctuations or pits in it. However, as shown in Figure 8 there are two significant dips, these are at final consensus clustering 7 and 9. This means that the number cuts for these two clusterings were considerably smaller than the cuts made in earlier and in later clusterings. This then leads us to the conclusion that both of these clusterings show the most stability in their partitions.

We can see from the results that the final clustering of 9 from the CBGF consensus function is the most consistent at gaining the highest values in terms of all our measurements. Though, other clusters have also presented comparatively good results, especially within both the CBGF and KMCF consensus functions the clustering of 9 appeared to be the most consistent overall.

Finally, it is worth highlighting that the work undertaken in Section 3.3 found that a good clustering, (a good balance between objective function values, tolerances and the number of clusters) of the data set was achieved with 9 clusters. Refer to Figures 1 and 2.

8 Conclusion

Phishing is carried out by multiple groups of people over the internet. In this study we were provided with the artefacts of phishing attacks on financial institutions in Australia by a major Australian Bank. The artefacts of this phishing activity are emails that have been identified and classified as phishing emails. This work has used different clustering techniques to identify the groups involved in phishing. The main problems with emails is how to represent them as objects that can be clustered. Our approach has been to use three different representations of the emails, text content as determined by words, link content as determined by the hidden links in the email and the orthographic structure as determined by the features in Section 3.3. These were all natural representations of the emails, however a fourth facet of phishing emails is the scripting, but this will be part of our future work.

The features from each of the three feature spaces mentioned above were selected and individual clustering algorithms were used to determine clusterings based on each of these representations. Each of these clusterings provided different information, not all suggested a number of phishing groups. However the orthographic approach using the Modified Global k-means algorithm and some analysis of the objective function (clustering function) suggested 9 groups.

In order to utilise the evidence from the three clustering approaches, they were ensembled using the three clustering consensus approaches as described in Section 4. Two of these graphing functions, CBGF and KMCF provided interesting results when the edge cut graphs were examined, again suggesting 9 as the likely number of final clusters. The NMI and purity measures between these consensus functions and the three clusterings of the text, links and orthographic techniques also demonstrated maximum mutual information and balance purity around 9 clusters. This can be clearly seen by the sum in Figures 9 and 10.

Whilst not conclusive, this paper has explored clustering approaches and ensemble clustering approaches to provide information about the number of phishing groups. This, through the different individual clustering representations provides information about the profile of these groups. This paper has concentrated on assembling the strongest evidence for identifying a specific number phishing groups.

The issue of identifying and articulating the profile of these particular groups will be the subject of a further paper. A model will be built using the clusterings found in this paper, where the separate information about the modus operandi of the groups can be brought together. Other future work will include a reality check of our results with expert views of the number and nature of phishing groups and testing our model on other data sets.

References

Alison, L., Smith, M., Eastman, O. & Rainbow, L. (2003), 'Toulmins philosophy of argument and its

relevance to offender profiling', Psychology, Crime and Law **9**(2), pp. 173–183.

Bagirov, A.M. & Mardaneh, K. (2006), Modified global k-means algorithm for clustering in gene expression data sets. In Proc. 2006 Workshop on Intelligent Systems for Bioinformatics (WISB 2006), vol. 73, Hobart, Australia. CRPIT.

Bagirov, A. M. (2008), Modified global k-means algorithm for minimum sum-of-squares clustering problems, Pattern Recogn, 41, 10 (Oct. 2008), pp. 3192–3199.

Bagirov, A.M. & Yearwood, J. (2006), A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. European Journal of Operatioanl Research, vol. 170 pp. 578–596.

Castle, T. & Hensley, C. (2002), 'Serial killers with military experience: Applying learning theory to serial murder, International Journal of Offender Therapy and Comparative Criminology pp. 453–465.

Chandrasekaran,M., Karayanan, K. & Upadhyaya, S. (2006), Towards phishing e-mail detection based on their structural properties, in In New York State Cyber Security Conference.

Chau, D. (2005), Prototyping a lightweight trust architecture to fight phishing, Technical report, MIT ComputerScience And Artificial Intelligence Laboratory. Final Report, **URL:** *http://groups.csail.mit.edu/cis/crypto/projects/antiphishing/*

Clark, R. (1993), Profiling: A hidden challenge to the regulation of data surveillance, Journal of Law and Information Science 4,2.

Emigh, A. (2005), Online identity theft: Phishing technology, chokepoints and countermeasures, Technical report, Radix Labs. Retrieved from Anti-Phishing Working Group:
URL: *http://www.antiphishing.org/resources.html*

Fern, X. Z. & Brodley C. E. (2004), Cluster Ensembles for High Dimensional Clustering: An Empirical Study, Journal of Machine Learning Research.

Fern, X. Z. & Brodley C. E. (2004), Solving cluster ensemble problems by bipartite graph partitioning, In Proceedings of the Twenty-First international Conference on Machine Learning (Banff, Alberta, Canada, July 04 - 08, 2004). ICML '04, Vol. 69. ACM, New York, NY, p. 36.

Fette, I., Sadeh, N. & Tomasic, A. (2007), Learning to detect phishing emails, in WWW 07: Proceedings of the 16th international conference on WorldWide Web, ACM Press, New York, NY, USA, pp. 649–656.

Freund, Y. & Schapire, R. (1999), 'A short introduction to boosting, Journal of Japanese Society for Artificial Intelligence **14**(5).

Thorsten, J. (2002), Learning to classify text using support vector machines: methods, theory and algorithms, Kluwer Academic Publishers, Norwell, MA, USA.

Jakobsson, M. & Young, A. (2005), 'Distributed phishing attacks, Cryptology ePrint Archive, Report 2005/091. **URL:** *http://eprint.iacr.org/*

Juels, A., Jakobsson M. & Jagatic, T. N. (2006), Cache cookies for browser authentication (extended abstract), in SP 06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P06), IEEE Computer Society, Washington, DC, USA, pp. 301–305.

Karypis, G. & Kumar V. (1998), A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.

Cover, T. M. & Thomas, J. A. (1991), *Elements of Information Theory.*, Wiley, 1991.

Fern, X. Z. & Brodley, C. E.(2004), 'Cluster ensembles for high dimensional clustering: An empirical study', *Journal of Machine Learning Research.*

Karypis, G. & Kumar, V. (1998), METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Centre, Minneapolis.

Topchy A., Jain A. K. & Punch, W. (2003), Combining multiple weak clusterings, *in* 'IEEE International Conference on Data Mining', pp. 331–338.

Strehl, A. & Ghosh, J. (2002), 'Cluster ensembles - A knowledge reuse framework for combining multiple partitions', *Journal of Machine Learning Research* 3, pp. 583–617.

Karypis, G. & Kumar, V. (1998), 'IEEE/ACM Conference on Supercomputing' SC98, 07-13 Nov. 1998 p. 28

Roger, C. (1993), 'Profiling: A Hidden Challenge to the regulation of Data Surveillance', *Journal of Law and Information Science*, 1993 **4**(2)

Webb, D. (2007), 'A Free and Comprehensive Guide to the World of Forensic Psychology', 'All about Forensic Psychology', **URL:** *http://www.all-about-forensic-psychology.com*

Wu, M., Miller, R. C. & Garfinkel, S. L. (2006), 'Do security toolbars actually prevent phishing attacks?', *in* ' Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montral, Qubec, Canada, April 22 - 27, 2006)'. CHI '06. ACM, NY, pp. 601–610.